

## Precaution of NX1 Using XIP and FIFO

**Description:** When IC of NX1\_OTP series is in the process of XIP and reading FIFO Data of the peripheral functions at the same time, the FIFO Data may be wrong.

**Reason:** NX1 series can execute the programs in SPI Flash through XIP (**eX**ecute **I**n **P**lace). Reading FIFO by executing XIP program, or reading FIFO by other process (e.g. ISR) while XIP is performed, the following peripheral functions of FIFO Data will be unexpected because FIFO is accessed by the other program at the same time.

- ADC FIFO
- UART FIFO
- SPI0 FIFO
- SPI1 FIFO

**Solution:** The error reading of relevant FIFO Data can be avoided by using other registers. The following provides the methods of using the relevant FIFO registers, with practical examples to illustrate the correct way to read FIFO Data. These example programs should not be executed through XIP.

- ADC FIFO

● **P\_ADC\_Data\_CH0 (ADC Channel 0 Register)**

Register	Offset	Description	Initial Value	
P_ADC_Data_CH0	ADC_BA+0x10	ADC Channel 0 Register	0x0000_0000	
Bit	Name	R/W	Descriptions	Initial Value
Bit[31:12]	Reserved			0x0
Bit[15:4]	CH0_DAT	R	ADC's Channel 0 Register	0x0
Bit[3:0]	Reserved			0x0

```
#include "nx1_lib.h"
#include "nx1_adc.h"
U16 getAdcFifo(void) {
    //Step.1 Read register, you can use it or discard.
    //      Use P_ADC->Ctrl for example.
    (void)P_ADC->Ctrl;
    //Step.2 Read ADC FIFO.
    return P_ADC->Data_CH0;
}
```

- UART FIFO

● **P\_UART\_Data (UART Data Register)**

Register	Offset	Description	Initial Value	
P_UART_Data	UART_BA+0x00	UART Data Register	0x0000_0000	
Bit	Name	R/W	Descriptions	Initial Value
Bit[31:0]	Reserved			
Bit[7:0]	UDR	R/W	write to TXD FIFO in WR (THR) read from RXD FIFO in RD (RBR)	0x0

```
#include "nx1_lib.h"
#include "nx1_uart.h"
U8 getUartFifo(void) {
    //Step.1 Read register, you can use it or discard.
```

```

//      Use P_UART->Ctrl for example.
(void)P_UART->Ctrl;
//Step.2 Read UART FIFO.
return P_UART->Data;
}

```

- SPI0 FIFO

- P\_SPI0\_Data (SPI0 Data Register)

Register	Offset	Description	Initial Value	
P_SPI0_Data	SPI_BA+0x10	SPI0 Data Register	0x0000_0000	
Bit	Name	R/W	Descriptions	Initial Value
Bit[31:0]	FIFO_D	R/W	Write Data into FIFO Read Data from FIFO	0x0

```

#include "nx1_lib.h"
#include "nx1_spi.h"
U32 getSpi0Fifo(void) {
//Step.1 Read register, you can use it or discard.
//      Use P_SPI0->Ctrl for example.
(void)P_SPI0->Ctrl;
//Step.2 Read SPI0 FIFO.
return P_SPI0->Data;
}

```

- SPI1 FIFO

- P\_SPI1\_Data (SPI1 Data Register)

Register	Offset	Description	Initial Value	
P_SPI1_Data	SPI_BA+0x50	SPI1 Data Register	0x0000_0000	
Bit	Name	R/W	Descriptions	Initial Value
Bit[31:0]	FIFO_D	R/W	Write Data into FIFO Read Data from FIFO	0x0

```

#include "nx1_lib.h"
#include "nx1_spi.h"
U32 getSpi1Fifo(void) {
//Step.1 Read register, you can use it or discard.
//      Use P_SPI1->Ctrl for example.
(void)P_SPI1->Ctrl;
//Step.2 Read SPI1 FIFO.
return P_SPI1->Data;
}

```